

Esercitazione

Un libro implementato come
astrazione di dato e tipo di dato
astratto in C

Traccia

- Implementare un modulo software che rappresenti un libro di una biblioteca
- Implementarlo con il linguaggio C come:
 - Astrazione di dato
 - Tipo di dato astratto

Caratteristiche

- Attributi:
 - Autore
 - Titolo
 - Numero di copie disponibili
- Operazioni:
 - Inizializzazione
 - Prestito
 - Restituzione

Astrazione di dato – libro.h

```
/* File libro.h
```

```
Questo file header dichiara TUTTE e SOLE le funzioni  
che possono essere usate per interagire con il libro  
Solo le dichiarazioni presenti in questo file header  
sono visibili "all'esterno" (ossia dagli altri punti del programma)
```

```
*/
```

```
void init(char* autore, char* titolo, int copie);
```

```
int disponibili();
```

```
void presta();
```

```
void restituisci();
```

```
void stampa();
```

Astrazione di dato – libro.c (1)

```
/* File: libro.c
```

Questo file definisce un modulo "libro", in modo che sia possibile interagire con il libro usando solo determinate operazioni

```
*/
```

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include "libro.h"
```

Astrazione di dato – libro.c (2)

```
/*  
Struttura che definisce il libro  
Siccome viene dichiarata "static", puo' essere vista solo all'interno di questo file =>  
e' protetta a livello di file  
Grazie a questo livello di protezione e' possibile definire funzioni/operazioni  
pubbliche che agiscano sul dato protetto  
garantendo cosı la coerenza dei dati  
*/  
  
static struct  
{  
    char autore[256]; /* autore del libro */  
    char titolo[256]; /* titolo del libro */  
    int copie; /* copie totali */  
    int disponibili; /* numero di copie disponibili */  
} libro;
```

Astrazione di dato – libro.c (3)

```
/* operazioni pubbliche */
```

```
/* inizializza il libro */
```

```
void init(char* autore, char* titolo, int copie)
```

```
{
```

```
    /* inizializzo gli attributi */
```

```
    strcpy(libro.autore, autore);
```

```
    strcpy(libro.titolo, titolo);
```

```
    /* all'inizio, tutte le copie sono disponibili */
```

```
    libro.copie = copie;
```

```
    libro.disponibili = copie;
```

```
}
```

```
/* restituisce il numero di copie disponibili */
```

```
int disponibili()
```

```
{
```

```
    return libro.disponibili;
```

```
}
```

Astrazione di dato – libro.c (4)

```
/* presta un libro */  
void presta()  
{  
    /* controllo che ci siano copie disponibili */  
    if (libro.disponibili > 0)  
        /* in caso positivo, decremento */  
        libro.disponibili--;  
    else  
        /* altrimenti do un messaggio di errore  
        e non decremento */  
        printf("Errore: non ci sono copie disponibili\n");  
}
```


Astrazione di dato – libro.c (5)

```
/* restituisce un libro */
void restituisci()
{
    /* controllo di non superare il numero di copie */
    if (libro.disponibili < libro.copie)
        /* in caso positivo, incremento le copie disponibili */
        libro.disponibili++;
    else
        /* altrimenti c'e' qualcosa che non va */
        printf("C'\e' qualcosa che non va...\n");
}

/* stampa le info sul libro */
void stampa()
{
    printf("Titolo: %s; autore: %s;\n", libro.titolo, libro.autore);
}
```

Astrazione di dato – mainDA.c (1)

```
/* File mainda.c
   Esempio di uso di libro.c
*/

#include <stdio.h>
#include "libro.h"

int main()
{
    /* inizializzazione */
    init("Collodi", "Pinocchio", 3);
    /* abbiamo solo una istanza di libro */
    stampa();
    printf("Copie disponibili: %d\n", disponibili());
    printf("Chiedo un libro in prestito\n");
    presta();
    printf("Ora la disponibilita\' e\' di %d copie\n", disponibili());
}
```

Astrazione di dato – mainDA.c (2)

```
presta();
presta();
printf("Dopo altri due prestiti la disponibilit  e\' %d\n", disponibili());
printf("Se provo a chiedere il libro in prestito ottengo:\n");
presta();
printf("Meglio restituire...\n");
restituisci();
restituisci();
restituisci();
printf("Ci sono %d copie: proviamo a restituire una copia di un altro libro...\n",
       disponibili());
restituisci();
printf("\n");
return 0;
}
```

Tipo di dato astratto – libroADT.h (1)

```
/* File libro.h
```

```
Questo file header dichiara un tipo di dato astratto "libro",  
e le operazioni che possono essere usate per interagire con il libro
```

```
*/
```

```
/*
```

```
Dichiarazione del tipo struttura che implementa il libro
```

```
*/
```

```
typedef struct
```

```
{
```

```
    char autore[256]; /* autore del libro */
```

```
    char titolo[256]; /* titolo del libro */
```

```
    int copie; /* copie totali */
```

```
    int disponibili; /* numero di copie disponibili */
```

```
} libro;
```

Tipo di dato astratto – libroADT.h (2)

```
/* Dichiarazione delle funzioni che agiranno  
   sul tipo di dato astratto libro  
   Notare che e' richiesto un puntatore all'istanza del libro  
*/
```

```
void init(libro* l, char* autore, char* titolo, int copie);  
int  disponibili(libro l);  
void presta(libro* l);  
void restituisci(libro* l);  
void stampa(libro l);
```

Tipo di dato astratto – libroADT.c (1)

```
/* File: libroADT.c
   Questo file definisce le operazioni del tipo di dato astratto "libro"
*/
#include <stdio.h>
#include <string.h>
#include "libroADT.h"

/* operazioni pubbliche */

/* inizializza il libro */
void init(libro* l, char* autore, char* titolo, int copie)
{
    /* inizializzo gli attributi */
    strcpy(l->autore, autore);
    strcpy(l->titolo, titolo);
    /* all'inizio, tutte le copie sono disponibili */
    l->copie = copie;
    l->disponibili = copie;
}
```

Tipo di dato astratto – libroADT.c (2)

```
/* restituisce il numero di copie disponibili */
/* non dovendo modificare il libro, non c'e' bisogno del puntatore */
int disponibili(libro l)
{
    return l.disponibili;
}

/* presta un libro */
void presta(libro* l)
{
    /* controllo che ci siano copie disponibili */
    if (l->disponibili > 0)
        /* in caso positivo, decremento */
        l->disponibili--;
    else
        /* altrimenti do un messaggio di errore
        e non decremento */
        printf("Errore: non ci sono copie disponibili\n");
}
```

Tipo di dato astratto – libroADT.c (3)

```
/* restituisce un libro */
void restituisci(libro* l)
{
    /* controllo di non superare il numero di copie */
    if (l->disponibili < l->copie)
        /* in caso positivo, incremento le copie disponibili */
        l->disponibili++;
    else
        /* altrimenti c'e' qualcosa che non va */
        printf("C'\e' qualcosa che non va...\n");
}

/* stampa le info sul libro */
void stampa(libro l)
{
    printf("Titolo: %s; autore: %s;\n", l.titolo, l.autore);
}
```


Tipo di dato astratto – mainADT.c (1)

```
/* File mainADT.c
   Esempio di uso di libroADT.c
*/
#include <stdio.h>
#include "libroADT.h"

int main()
{
    libro l1, l2; /* abbiamo due istanze di libro */
    /* inizializzazione */
    init(&l1, "Collodi", "Pinocchio", 2);
    init(&l2, "Manzoni", "I promessi sposi", 3);
    /* usiamo le istanze */
    printf("libro 1:\n");
    stampa(l1);
    printf("Copie disponibili: %d\n", disponibili(l1));
    printf("libro 2:\n");
    stampa(l2);
    printf("Copie disponibili: %d\n", disponibili(l2));
}
```

Tipo di dato astratto – mainADT.c (2)

```
printf("Chiedo una copia di Pinocchio in prestito\n");
presta(&l1);
printf("Ora la disponibilita\' e\' di %d copie\n", disponibili(l1));
printf("Mentre la disponibilita\' de l promessi sposi rimane %d\n", disponibili(l2));
presta(&l1);
printf("Dopo un altro prestito la disponibilita\' e\' %d\n", disponibili(l1));
printf("Se provo a chiedere il libro in prestito ottengo:\n");
presta(&l1);
printf("Mentre posso prendere una copia de l promessi sposi\n");
presta(&l2);
printf("Adesso ne rimangono %d copie\n", disponibili(l2));
printf("Meglio restituire Pinocchio...\n");
restituisci(&l1);
restituisci(&l1);
printf("Ci sono %d copie di Pinocchio: proviamo a restituire un\'altra copia...\n", disponibili(l1));
restituisci(&l1);
printf("\n");
return 0;
}
```